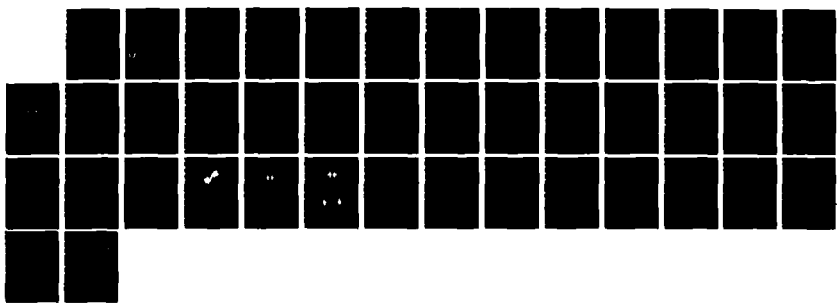
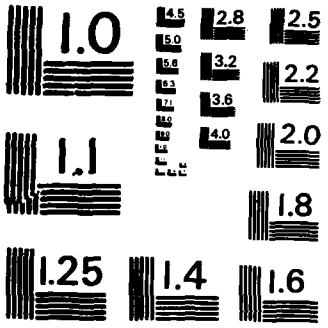
AD-A156 663 A MESH MOVING TECHNIQUE FOR TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS (U) ARMY ARMAMENT RESEARCH AND DEVELOPMENT CENTER WATERVLIET NY L D C ARNEY ET AL 1/1

UNCLASSIFIED JUN 85 ARLCB-TR-85016 SBI-AD-E440 287 F/G 12/1 NL





AD

TECHNICAL REPORT ARLCB TR 85016

A MESH MOVING TECHNIQUE FOR TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS IN TWO SPACE DIMENSIONS

DAVID C. ARNEY

JOSEPH E. FLAHERTY

JUNE 1985

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT CENTER
— LARGE CALIBER WEAPON SYSTEMS LABORATORY —
BENET WEAPONS LABORATORY
WATERVLIET N.Y. 12189

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The use of trade name(s) and/or manufacture(s) does not constitute an official indorsement or approval.

DISPOSITION

Destroy this report when it is no longer needed. Do not return it to the originator.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ARLCB-TR-85016	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A MESH MOVING TECHNIQUE FOR TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS IN TWO SPACE DIMENSIONS	5. TYPE OF REPORT & PERIOD COVERED: Final
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): David C. Arney and Joseph E. Flaherty (see reverse)	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: US Army Armament Research & Development Center, Benet Weapons Laboratory, SMCAR-LCB-TL, Watervliet, NY 12189-5000	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: AMCMS No. 6111.01.011; PRON No. 1A425M51A1A
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Armament Research & Development Center, Large Caliber Weapon Systems Laboratory, Dover, NJ 07801-5001	12. REPORT DATE: June 1985
	13. NUMBER OF PAGES: 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
Presented at the Second Army Conference on Applied Math & Computing, RPI, 22-24 May 1984. The authors were partially supported by the U.S. Army Research Office under Contract Number DAAG-82-K-0197 and the U.S. Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant Number AFOSR 80-0192.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Adaptive Methods
Hyperbolic Partial Differential Equations
Mesh Moving
Finite Volume

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
We discuss an adaptive mesh moving technique that can be used with a finite difference or finite element scheme to solve initial-boundary value problems for vector systems or partial differential equations in two space dimensions and time. The mesh moving technique is based on an algebraic node movement function determined from the propagation of significant error regions. The algorithm is designed to be flexible, so that it can be used with many existing
(CONT'D ON REVERSE)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

7. AUTHOR(S) (CONT'D)

David C. Arney
Department of Mathematics
United States Military Academy
West Point, NY 10996
and
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12181

Joseph E. Flaherty
Armament Research and Development Center
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189-5000
and
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12181

20. ABSTRACT (CONT'D)

finite difference and finite element methods. To test the algorithm, we implemented the mesh mover in a system code along with an initial mesh generator and a MacCormack finite volume integrator on quadralateral cells to solve hyperbolic vector systems. Results are presented for several computational examples. This moving mesh reduces dispersion errors near shocks and wave fronts and thereby reduces the grid requirements necessary to compute accurate solutions while increasing efficiency.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

Mesh moving is an adaptive technique that has been used successfully to improve the accuracy of both finite element and finite difference schemes for a variety of time dependent problems in one space dimension (cf., e.g., [1,2,10,11,14,18,20,23]).[1] The essential idea is to derive equations so that the mesh moves either to extremize some quantity, e.g., to minimize the discretization error, or to follow some local nonuniformity, e.g., a wave front. This generally reduces dispersive errors and allows the use of larger time steps while maintaining accuracy. For example, with a fixed mesh a wave front may move through a cell in one time step causing significant dispersive errors (cf. Figure 1a); whereas, a moving mesh with the same time step can follow the wave front and keep it within the same cell (cf. Figure 1b).

Mesh moving algorithms have often been related to the numerical integration scheme and/or the problem being solved. In one dimension, for example, Hyman [19] moves a mesh to minimize the time variation of the solution at the nodes. This scheme uses finite difference approximations for solving hyperbolic conservation laws. Davis and Flaherty [9] develop a finite element code for parabolic systems that moves the mesh so as to equidistribute the spatial component of the discretization error. Miller et al. [15,20,21] couple the node position equations into the finite element variational equations and minimize the

---

[1] references are listed at the end of the report

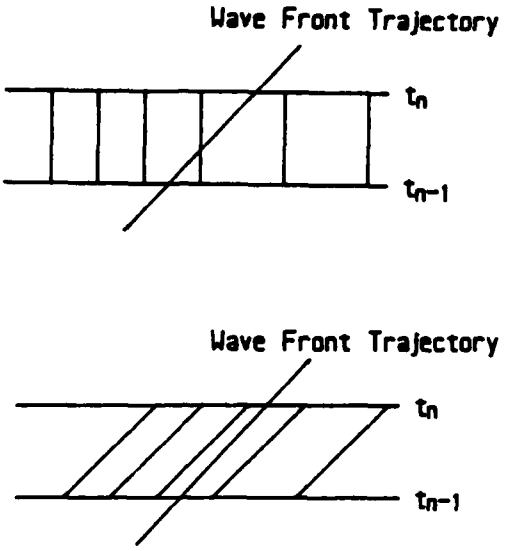


Figure 1. Wave Front Trajectory on a) Stationary Mesh (top) and b) Moving Mesh (bottom).

residual in solving parabolic problems. Bell and Shubin [2] solve the Euler-Lagrange equations of an extremizing functional and use a finite difference scheme to solve hyperbolic conservation laws. All of these schemes have successfully demonstrated that mesh moving can reduce error and provide improvements in computational efficiency for one-dimensional problems.

With some modification the Hyman and Miller algorithms can be extended to higher dimensions; however, many other mesh moving techniques are not directly applicable to two- and three-dimensional problems. One difficulty is that equidistribution strategies fail to produce unique solutions. Brackbill and Saltzman [7,26] have overcome

this problem by adding the constraints of mesh smoothness and orthogonality to a variational problem.

A sucessful mesh moving scheme for higher dimensional problems that is somewhat similar to the method presented here is the algorithm of Rai and Anderson [23,24,25]. Their algorithm is based on a gravitional principle and calculates the velocity of a node based on the difference between its error and the mean error. The displacement of one node with respect to another is inversely proportional to the distance between them. A summation over all nodes is necessary to determine each node's speed in a computational grid.

A different adaptive technique is local mesh refinement which consists of dividing or refining elements in regions where the solution is not adequately resolved. The advantage of this technique relative to mesh moving is that enough fine grids can be added to resolve the small scale structures of the solution and provide solutions to within user prescribed error tolerances. The local mesh refinement schemes of Berger [3,4], Flaherty and Moore [13], Gannon [16] and Bieterman and Babuska [5,6], have successfully satisfied user tolerances for different problems using finite element or finite difference schemes in either one or two dimensions.

The most promising algorithms appear to be those that combine both mesh movement and local mesh refinement. While neither adaptive technique or their combination is likely to be optimal for most problems, a combination can accurately solve for the solution in regions

where it varies rapidly and devote little effort in regions where it varies slowly. It is our intention to consider such schemes; however, the computational procedures discussed here do not as yet contain local refinement.

The mesh moving scheme we have developed is simple, efficient, and independent of the numerical method being employed to discretize the partial differential equations. At each time step it uses the current node locations and the nodal values of a mesh movement indicator. We use local error estimates as mesh movement indicators, but other computable values such as solution gradients or curvature could be used. Nodes with "statistically significant error" (cf. Section 2) are grouped into rectangular error clusters. This clustering separates spatially distinct phenomena of the solution. As time evolves the clusters can move, change size, change orientation, collide, separate, reflect off boundaries, or pass through boundaries. At each time step new clusters can be created, and old ones can vanish. The clustering algorithms we use are briefly described in Section 2 and were developed by Berger [3,4] for a mesh refinement scheme for solving hyperbolic problems.

Mesh movement is determined by the node's relationship to its nearest error cluster. Movement is done in two steps, each in a direction along a principal axis of a cluster rectangle. The amount of movement in each direction is determined by a movement function which insures that the center of error of the cluster moves according to a

differential equation suggested by Coyle et al. [8]. Additionally, the movement function smoothes mesh motion, reduces distortion, mesh tangling, or overlapping, and prevents nodes from moving outside the domain boundaries.

In Section 2 we discuss error clustering, movement of the center of mass of the error cluster, the node movement function, and the initial mesh generator used in the computational examples. In Section 3 we discuss the MacCormack finite volume scheme for hyperbolic equations and the error estimates used in the computational examples. The results of the computational examples are given in Section 4, and Section 5 contains a discussion of the results of the experiments and the status of our algorithm.

2. MESH MOVING SCHEME AND INITIAL MESH GENERATION

We suppose that an approximate solution of the partial differential equation and a pointwise error estimate have been calculated by some numerical technique at the current time step. We then flag "significantly high error nodes" as nodes with error greater than twice the mean nodal error and also greater than a user-supplied error tolerance. If there are no significant error nodes, computation is performed on a stationary mesh. Next the nearest neighbor clustering algorithm of Berger [3,4] is used to cluster flagged error nodes. The nearest neighbor clusters have internodal distances in the cluster less than intercluster distances, which are the minimum distances between clusters. The formation of a cluster is done iteratively by starting

with a node and including nodes in a cluster if the distance from the node to the cluster is less than a specified distance. When a node is determined to belong to two or more clusters, those clusters are merged.

Berger [3,4] shows that near minimum area rectangles that contain all the nodes within the cluster can be easily generated. The principal axes of such a rectangle are the major and minor axes of an enclosed ellipse with the same first and second moments as the clustered nodes. Thus, if x_m and y_m are the mean coordinates of the clustered nodes, then the axes of the rectangle are the eigenvectors of the symmetric (2x2) matrix

$$\begin{bmatrix} \Sigma\, x_i^2 - x_m^2 & \Sigma\, x_i\, y_i - x_m y_m \\ \Sigma\, x_i y_i - x_m\, y_m & \Sigma\, y_i^2 - y_m^2 \end{bmatrix} \qquad (2.1)$$

For problems with significant error nodes located on a long curved line, the entire region will belong to one unacceptably large cluster. In order to prevent this inefficiency and provide better alignment with curved fronts, the rectangular clusters are checked for efficiency by determining the percentage of flagged nodes in the cluster to the total nodes in the cluster. If a 50 percent efficiency is not achieved, the rectangle is iteratively bisected in the direction of the major axis. This is repeated until all clusters have a 50 percent

efficiency or more. This nearest neighbor clustering separates spatially distinct phenomena as shown by the dotted clusters on the two dimensional mesh of Figure 2 and provides some linear alignment with long curved gradient fronts as shown by the clusters in Figure 3.

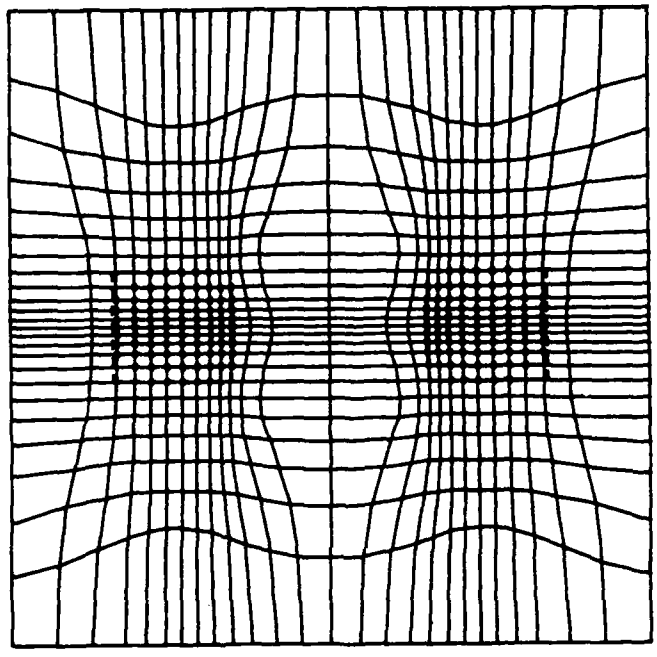


Figure 2. Two spatially distinct Clusters.

In order to determine proper node movement, as shown for a one dimensional problem in Figure 1b, the speed of propagation of the error clusters must be determined. Hyman [19] and Hyman and Harten [18] move nodes based on minimizing the time variation of the solution components. For hyperbolic systems this allows the mesh to move at a weighted average of the characteristic speeds at the node. Front tracking schemes also move the mesh so that isolated

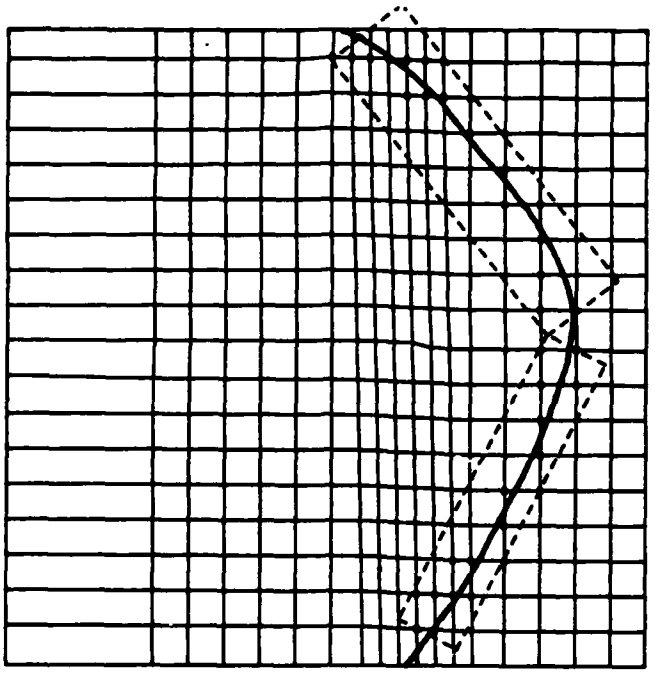


Figure 3. Clusters aligned with a Curved Front.

discontinuities are stationary in reference to the mesh. Since tracking error is possible for all time dependent problems, our approch is more general and is also an approximation to these schemes for hyperbolic problems, where error propagates in a characteristic direction. We assume that nodes in the same cluster have related solution characteristics, so that we can determine individual node movement from the propagation of the center of mass of the error cluster.

In the Hyman and Harten algorithm [18], when there is multiple wave interaction in a vector system, the best that can be done is to move the mesh with a weighted average of characteristic velocities. The same principle applies to our algorithm when multiple error clusters

have merged, the mesh is still able to move as their combined error cluster moves, which is a form of weighted averaging. Comparison between the center of mass propagation and the characteristic path is shown in Example 4.1 of Section 4.

We attempted to move nodes by a procedure that was based on extrapolating for the center of masses; however, this resulted in an unstable oscillatory effect. Indeed, Coyle et al. [8] showed that node movement based on extrapolation can be unstable. Using their suggestion, we stablize the movement by solving the differential equation

$$\ddot{r} + \lambda \dot{r} = 0 , \tag{2.2}$$

where r(t) is the position vector of the center of mass of an error cluster and $\dot{(\ )} := d(\ )/dt$. Equation (2.2) is conditionally stable, and when solved numerically with reasonable choices of $\lambda > 0$ the oscillations in the mesh movement were no longer present.

We solve Equation (2.2) from, say, t_n to t_{n+1} and hence, determine $r(t_{n+1})$ and the vector $r(t_{n+1}) - r(t_n)$ which is projected in the two axial directions of the rectangular cluster to determine the maximum movement (MM) in each axial direction. Along the two axial directions the movement function is one-dimensional. A profile of the node movement function that we use is shown in Figure 4; however, the algorithm is designed to be used with any general one-dimensional movement function. Slopes a and b depend on the distance from the

cluster to the domain boundaries and other clusters.

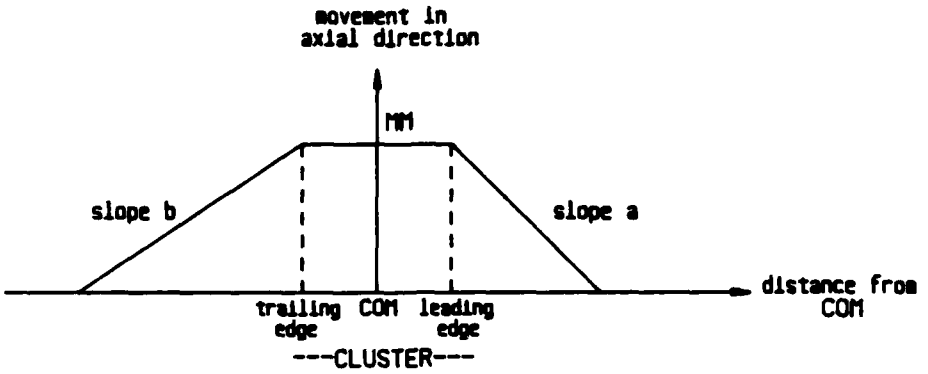


Figure 4. Profile of the Node Movement Function.

As shown in Figure 4, we let

$$MA_{inside} = \begin{cases} MM(1-ax) & \text{if nodes are x distance ahead of leading edge} \\ MM & \text{if nodes are inside the cluster projection} \\ MM(1-bx) & \text{if nodes are x distance behind trailing edge .} \end{cases} \quad (2.3)$$

Equation (2.3) determines the movement distance for nodes inside the projection of the cluster on the axis, i.e., the shaded region of Figure 5. In order to provide smooth node movement throughout the domain, nodes outside this region move in a reduced amount as determined by

$$MA_{outside} = MA_{inside}[\ 1-(2z/DIAM)]\ , \quad (2.4)$$

where z is the distance outside the cluster projection as shown in

Figure 5 and DIAM is the diameter of the domain.

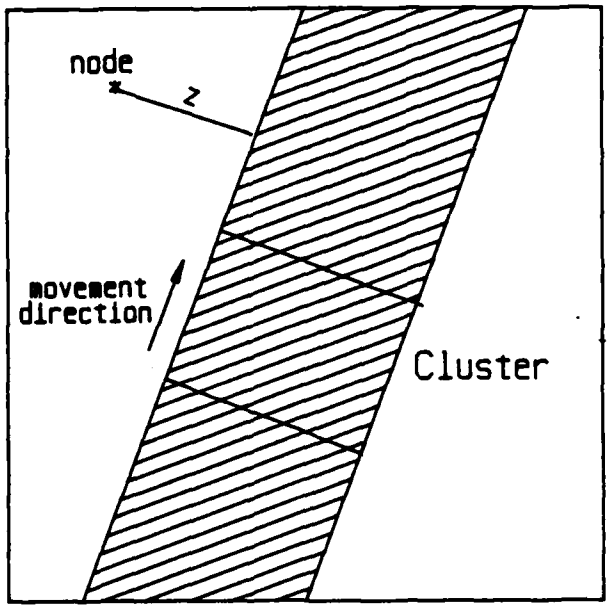


Figure 5. Nodes outside the Projection of the Error Cluster.

The generation of a proper initial mesh is critical to the success of the mesh moving scheme. Without refinement the mesh moving algorithm can not provide suitable error control, unless the initial mesh spacing properly resolves initial data. An initial error measure appropriate for the finite volume scheme on quadrilateral cells of Section 3 is the error in interpolating the prescribed initial condition $u_0(x,y)$ on each cell by a bilinear polynomial. The error on each cell is determined as the difference between the value of the initial function and its bilinear interpolant at the center of each cell. Therefore, the initial mesh must be generated so that the condition

$$|1/4\{u_0(x_i,y_i)+u_0(x_j,y_j)+u_0(x_k,y_k)+u_0(x_l,y_l)\}-u_0(\bar{x},\bar{y})| < TOL \quad (2.5)$$

holds on each cell when using the vertex and center point labelling as shown for a general cell in Figure 6. TOL is a user supplied error tolerance. We used the following iterative scheme to satisfy condition (2.5) for the computational examples of Section 4:

1. Input domain boundaries and initial data function.

2. Generate a uniform mesh.

3. Compute cell error from the left hand side of (2.5).

4. Cluster high error nodes and move influenced nodes toward center of clusters according to (2.3) and (2.4).

5. Smooth the mesh by solving the Euler-Lagrange equations of Brackbill and Saltzman [7].

6. Recompute error on cells.

Repeat

7.1. Add a mesh row and column to divide cells with error greater than TOL

7.2. Smooth the mesh by the algorithm of Brackbill and Saltzman[7]

7.3. Recompute the error

Until the error tolerance condition (2.5) is satisfied.

Initial meshes generated with this algorithm are shown in Figures 2 and 8 for the initial condition functions (4.6) and (4.2) respectively.

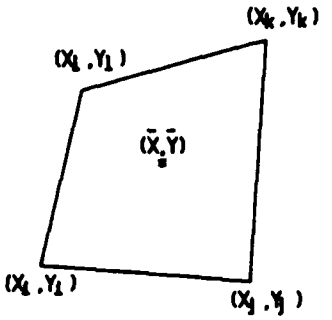


Figure 6. Cell Labelling for Equation (2.5).

3. MacCORMACK FINITE VOLUME SOLVER AND ERROR ESTIMATION

In order to test our mesh moving scheme, we used the explicit finite volume MacCormack scheme on *nonuniform* quadralateral grids for hyperbolic vector systems of the form

$$u_t + f_x(x,y,u,t) + g_y(x,y,u,t) = 0, \tag{3.1}$$

$$u(x,y,0) = u_0(x,y), \tag{3.2}$$

with appropriate well posed boundary conditions.

We index the nodes in a logically rectangular fashion where the time dependent node locations have cartesian coordinates $(x_{i,j}(t), y_{i,j}(t))$ and therefore, the time derivative of $u(x_{i,j}(t), y_{i,j}(t), t)$ is given by

$$\dot{u} = u_x \dot{x}_{i,j} + u_y \dot{y}_{i,j} + u_t \quad . \tag{3.3}$$

At time step n, we approximate

$$\dot{x}_{i,j}(t_n) \approx \Delta x_{i,j}(t_n)/\Delta t_n = (x_{i,j}^{n+1} - x_{i,j}^{n})/\Delta t_n \tag{3.4}$$

$$\dot{y}_{i,j}(t_n) \approx \Delta y_{i,j}(t_n)/\Delta t_n = (y_{i,j}^{n+1} - y_{i,j}^{n})/\Delta t_n \tag{3.5}$$

where $x_{i,j}^{n} = x_{i,j}(t_n)$, $y_{i,j}^{n} = y_{i,j}(t_n)$, and Δt_n is the current time step.

The finite volume scheme is obtained by integrating Equation (3.1) over each cell, where the general cell(i,j) with center $(x_{i,j}, y_{i,j})$ is shown in Figure 7. The area integrals involving the spatial derivatives of f and g are converted to line integrals of f and g around the cell boundaries using Green's Theorem. The integral of the time derivative term over cell(i,j) is approximated by

$$\iint u_t dxdy \approx A_{i,j}^{n} u_t \tag{3.6}$$

where $A_{i,j}^{n}$ is the area of cell(i,j) at timestep n. The line integrals are approximated by using values of the solution at nodes on appropriate sides of the cell boundaries. The predictor step of the MacCormack scheme uses node values to the left and below the boundaries, while the corrector uses node values to the right and above the boundaries.

After substitution of (3.4), (3.5), (3.6), and the appropriate line integral approximations into Equation (3.1) and using

$$F(u^n_{i,j}) = f(u^n_{i,j}) - (\Delta x_{i,j}(t_n)/\Delta t_n)u^n_{i,j} \tag{3.7}$$

$$G(u^n_{i,j}) = g(u^n_{i,j}) - (\Delta y_{i,j}(t_n)/\Delta t_n)u^n_{i,j} \tag{3.8}$$

where $u^n_{i,j}$ is the calculated approximation to $u_{i,j}(t_n)$ to simplify the function evaluations, the predictor and corrector Equations (3.9) and (3.10) for the MacCormack finite volume scheme on a moving mesh are obtained.

The predictor step is

$$\bar{u}^{n+1}_{i,j} = u^n_{i,j} - \Delta t_n/A^n_{i,j} \{ F(u^n_{i,j}) (y_{i+1/2,j+1/2} - y_{i+1/2,j-1/2}) - F(u^n_{i-1,j}) (y_{i-1/2,j+1/2} - y_{i-1/2,j-1/2}) + F(u^n_{i,j}) (y_{i-1/2,j+1/2} - y_{i+1/2,j+1/2}) - F(u^n_{i,j-1}) (y_{i-1/2,j-1/2} - y_{i+1/2,j-1/2}) - G(u^n_{i,j}) (x_{i+1/2,j+1/2} - x_{i+1/2,j-1/2}) + G(u^n_{i-1,j}) (x_{i-1/2,j+1/2} - x_{i-1/2,j-1/2}) - G(u^n_{i,j}) (x_{i-1/2,j+1/2} - x_{i+1/2,j+1/2}) + G(u^n_{i,j-1}) (x_{i-1/2,j-1/2} - x_{i+1/2,j-1/2})\}. \tag{3.9}$$

The corrector step is

$$u^{n+1}_{i,j} = 1/2 \{ u^n_{i,j} + \bar{u}^{n+1}_{i,j} - \Delta t_n/A^n_{i,j} [F(\bar{u}^{n+1}_{i+1,j}) (y_{i+1/2,j+1/2} - y_{i+1/2,j-1/2}) - F(\bar{u}^{n+1}_{i,j}) (y_{i-1/2,j+1/2} - y_{i-1/2,j-1/2}) + F(\bar{u}^{n+1}_{i,j+1}) (y_{i-1/2,j+1/2} - y_{i+1/2,j+1/2}) - F(\bar{u}^{n+1}_{i,j}) (y_{i-1/2,j-1/2} - y_{i+1/2,j-1/2}) - G(\bar{u}^{n+1}_{i+1,j}) (x_{i+1/2,j+1/2} - x_{i+1/2,j-1/2}) + G(\bar{u}^{n+1}_{i,j}) (x_{i-1/2,j+1/2} - x_{i-1/2,j-1/2}) - G(\bar{u}^{n+1}_{i,j+1}) (x_{i-1/2,j+1/2} - x_{i+1/2,j+1/2}) + G(\bar{u}^{n+1}_{i,j}) (x_{i-1/2,j-1/2} - x_{i+1/2,j-1/2})]\}. \tag{3.10}$$

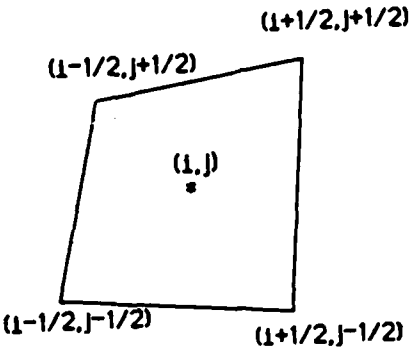


Figure 7. Labelling for general cell(i,j) for finite volume Equations (3.9) and (3.10).

On a stationary rectangular mesh, this scheme reduces to the standard MacCormack finite difference scheme, which when the predictor and corrector are combined for a linear partial differential equation is the same as the Lax-Wendroff finite difference scheme.

Accurate error estimation is important to insure that user tolerances are achieved and to refine proper regions when doing local mesh refinement. However, mesh moving is not as sensitive to error estimation. As long as the error estimator shows the error propagation, proper error magnitudes are not necessary. Therefore, in the computational examples of Section 4, we were able to use the difference between the predicted solution of Equation (3.9) and the corrected solution of Equation (3.10), as the error estimation or movement indicator. This error estimation is actually an estimation for the first order predicted solution, not the second order corrected solution, but does have the proper propagation characteristic.

A more accurate error estimation will be needed when local mesh refinement is implemented. Accurate error estimators that could be used

are Richardson extrapolation for finite difference schemes [3] and hierarchical methods for finite element schemes [28].

4. COMPUTATIONAL EXAMPLES

The following linear hyperbolic equations were solved as tests of our mesh moving technique. We used the initial mesh generator of Section 2 and the MacCormack solver described in Section 3. For each problem the two-dimensional domain was a square with sides of length 2 centered at the origin.

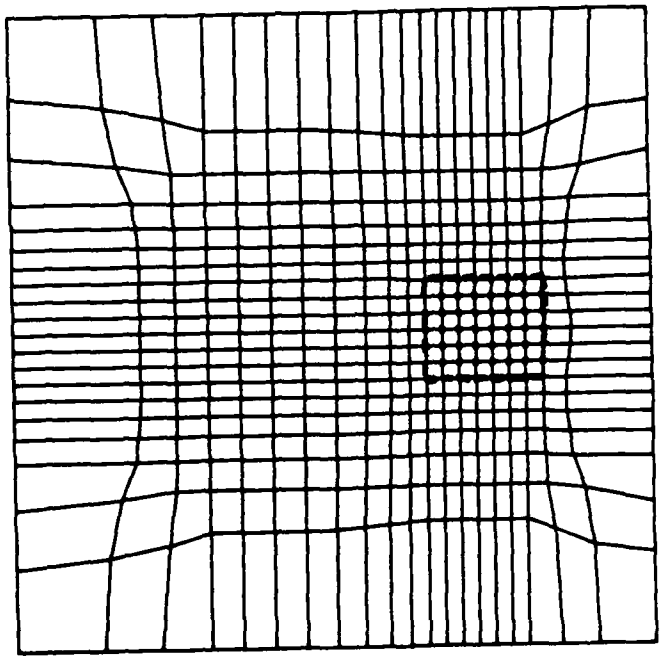


Figure 8. Initial Mesh for Example 4.1.

Example 4.1 Consider the initial-boundary value problem

$$u_t - yu_x + xu_y = 0, \qquad t > 0 \tag{4.1}$$

$$u(x,y,0) = \begin{cases} 0, & \text{if } (x-1/2)^2 + 1.5\,y^2 \geq 1/16 \\ 1-16((x-1/2)^2+1.5y^2), & \text{otherwise} \end{cases} \tag{4.2}$$

$$u(1,y,t) = u(-1,y,t) = u(x,-1,t) = u(x,1,t) = 0 \ . \tag{4.3}$$

The exact solution of this problem is

$$u(x,y,t) = \begin{cases} 0, & \text{if } C < 0 \\ C, & \text{if } C \geq 0, \end{cases} \tag{4.4}$$

$$C = 1-16((x\cos t + y\sin t - 1/2)^2 + 1.5(y\cos t - x\sin t)^2). \tag{4.5}$$

Equations (4.4) and (4.5) represent a moving elliptical cone rotating counterclockwise around the origin with period 2π. It was proposed as a test problem by Gottlieb and Orszag [17] and we selected it because the rotational quality of the the error region is a good test of a mesh moving scheme.

The initial mesh generated for this problem is shown in Figure 8. This mesh has an initial interpolation error less than 0.08. Figure 9 shows the mesh at t = 1.6, and Figure 10 shows the mesh at t = 3.2. The nodes follow the moving cone to keep it within the refined region. The dashed lines on Figures 8, 9, and 10 are the error cluster rectangles at the appropriate time steps. Figures 11 and 12 show the

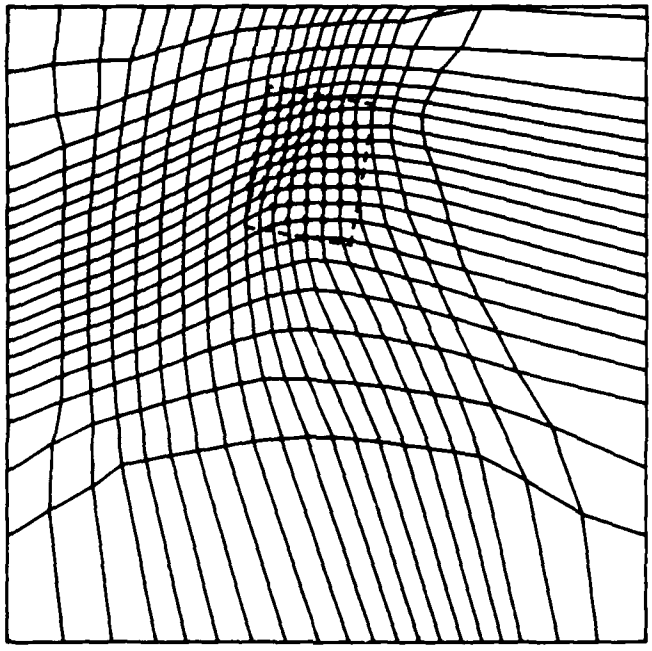


Figure 9. Mesh for Example 4.1 at t = 1.6

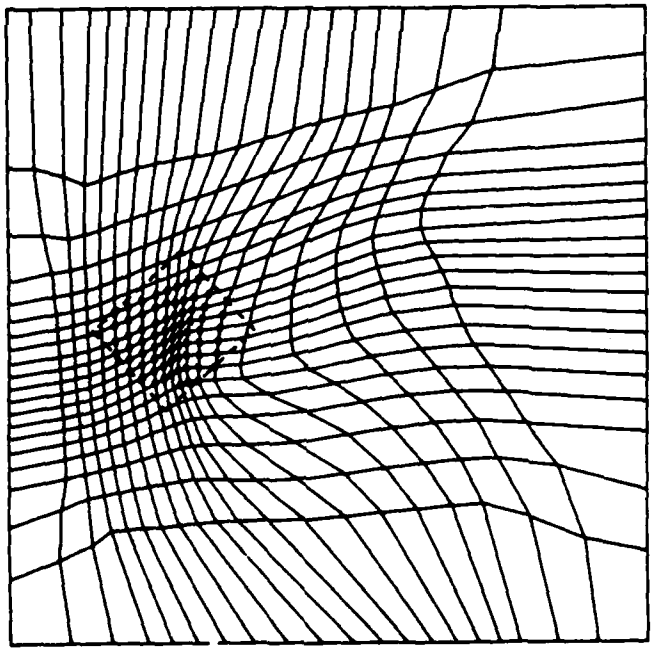


Figure 10. Mesh for Example 4.1 at t = 3.2

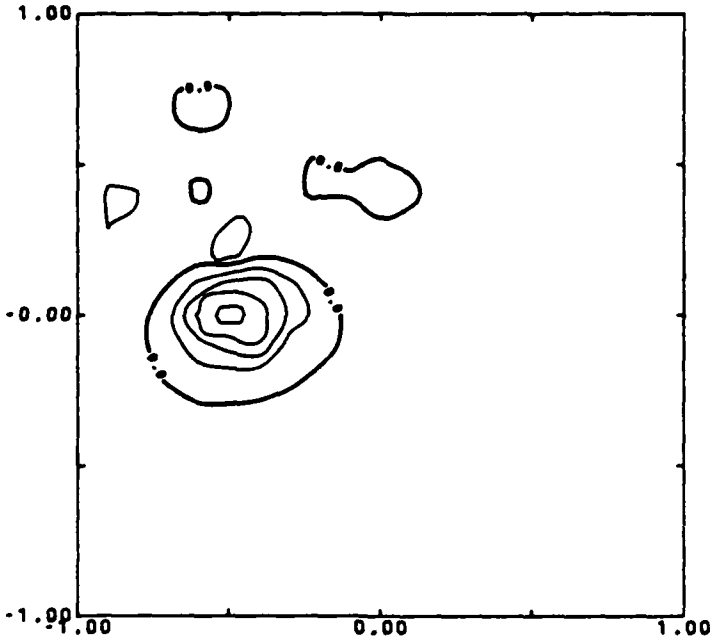


Figure 11. Contour Plot of Solution of Example 4.1 by MacCormack finite volume method on a moving mesh at t = 3.2

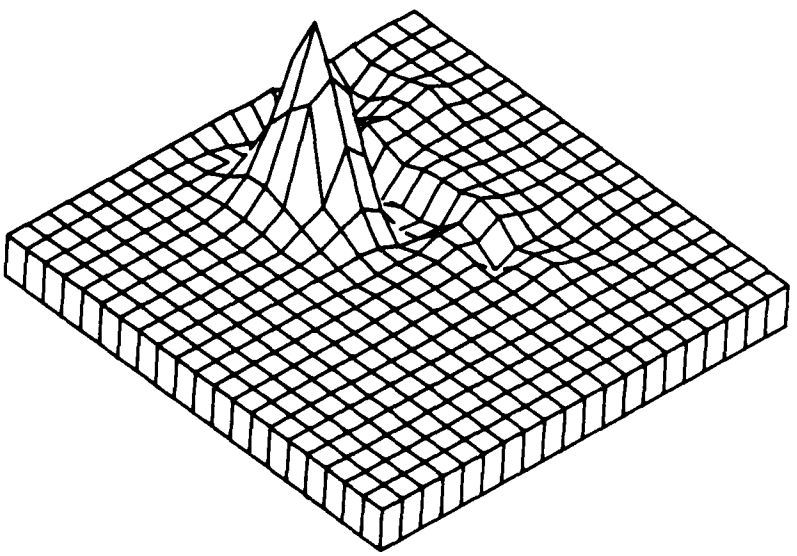


Figure 12. Surface Plot of Solution of Example 4.1 by MacCormack finite volume method on a moving mesh at t = 3.2

contour and surface plot of the solution at t = 3.2. The dispersion error in the form of a wake behind the cone for the moving mesh solution is reduced significantly from the wake in the solution using the Lax-Wendroff finite difference scheme on a 20x20 uniform stationary mesh. For comparison the contour plot and surface plot of the Lax-Wendroff solution at the same total time t = 3.2 and same number of time steps are shown in Figures 13 and 14. Figure 15 compares the path of the center of mass propagation using Equation (2.2) and the real characteristic path of the peak of the cone. As expected for this scalar hyperbolic problem, the vectors for the movement of the center of error mass determined by Equation (2.2) closely approximate the characteristic vectors of the center of the cone with a maximum difference of 15 percent in length and direction.

Example 4.2 This problem is a scalar double rotating cone problem with two symmetric cones rotating counterclockwise around the origin. The problem is given by Equations (4.1), (4.3), and new initial conditions provided by

$$u(x,y,0) = \begin{cases} 1-16((x-1/2)^2+1.5y^2), & \text{if } (x-1/2)^2+1.5y^2 \le 1/16 \\ 1-16((x+1/2)^2+1.5y^2), & \text{if } (x+1/2)^2+1.5y^2 \le 1/16 \\ 0, & \text{otherwise} \end{cases} \qquad (4.6)$$

Figure 16 shows the mesh at t = 1.15. Figure 16 shows the poor aspect ratio and mesh distortion caused by the rotation. The mesh

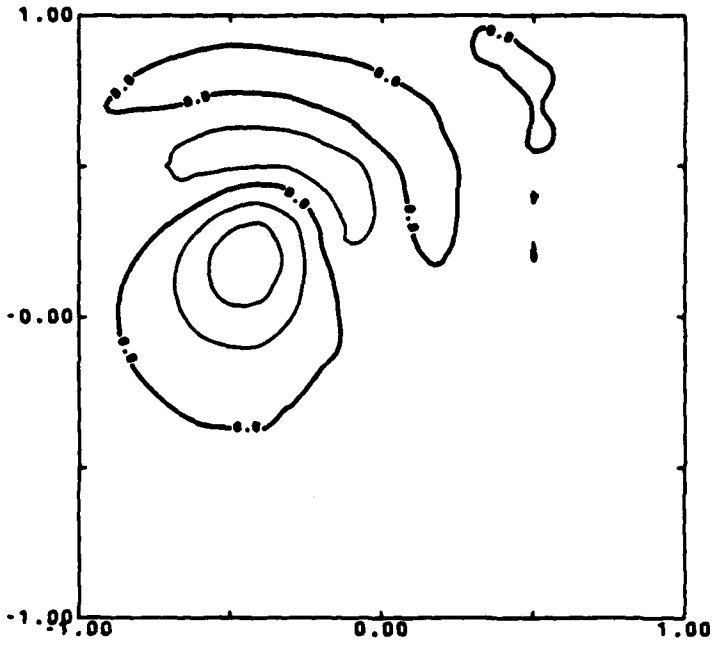


Figure 13. Contour Plot of Solution by the Lax-Wendroff scheme on a fixed 20x20 Mesh at t = 3.2.

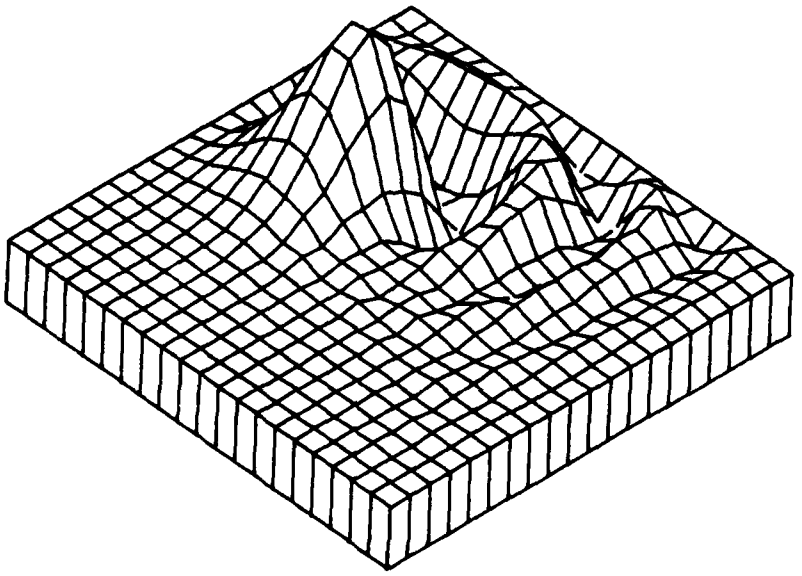


Figure 14. Surface Plot of Solution by the Lax-Wendroff scheme on a fixed 20x20 Mesh at t = 3.2.

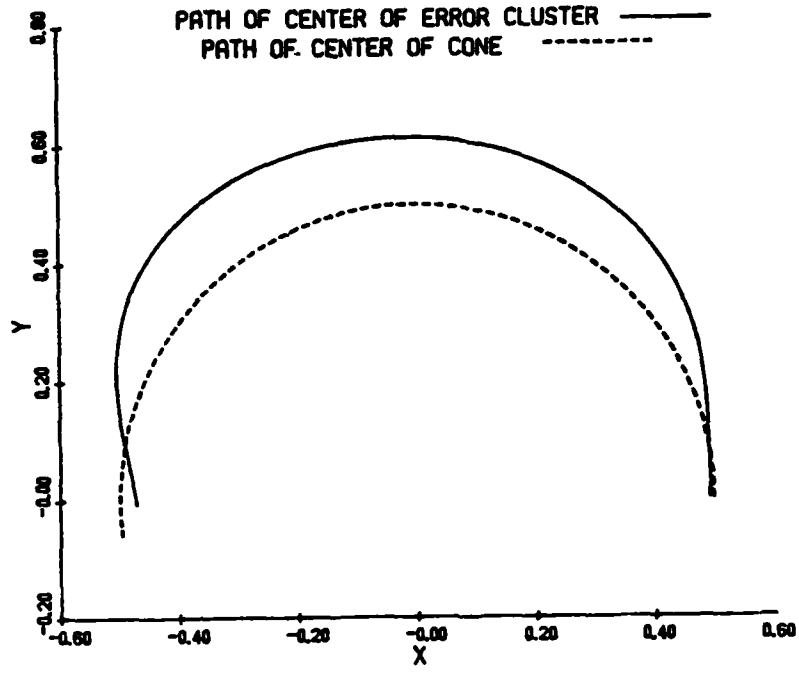


Figure 15. Comparison of characteristic path of center of cone and path of center of error mass as determined by Equation (2.2) for Example 4.1.

tangles as the cones rotate further. When mesh tangling occurs a static rezone that creates a new mesh using an algorithm similar to the one that generated the initial mesh must be employed. The data for the mesh can to be obtained by interpolation from the calculated solution at the nodes.

Example 4.3 This problem is an uncoupled system of moving cones that pass through one another. This causes the error clusters to collide and merge, and then later separate. The problem is given in Equations (4.7), (4.8) and (4.9).

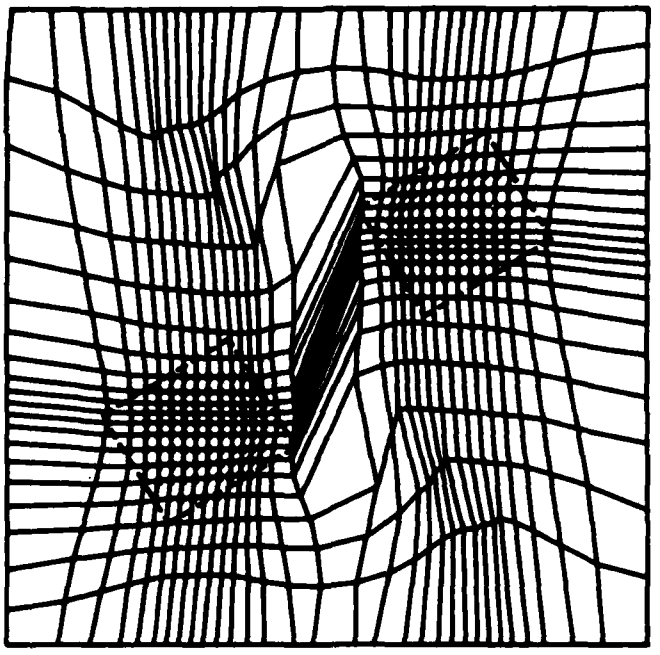


Figure 16. Distorted Mesh of Example 4.2 at t = 1.15.

$$u_t + u_x = 0 \tag{4.7a}$$

$$v_t - v_x = 0 \tag{4.7b}$$

$$v(x,y,0) = \begin{cases} 1-16((x+1/2)^2+1.5y^2), & \text{if } (x+1/2)^2+1.5y^2 \le 1/16 \\ 0, & \text{otherwise} \end{cases} \tag{4.8a}$$

$$u(x,y,0) = \begin{cases} 1-16((x-1/2)^2+1.5y^2), & \text{if } (x-1/2)^2+1.5y^2 \le 1/16 \\ 0, & \text{otherwise} \end{cases} \tag{4.8b}$$

$$u(x,y,t) = v(x,y,t) = 0 \text{ on all boundaries of the domain} \tag{4.9}$$

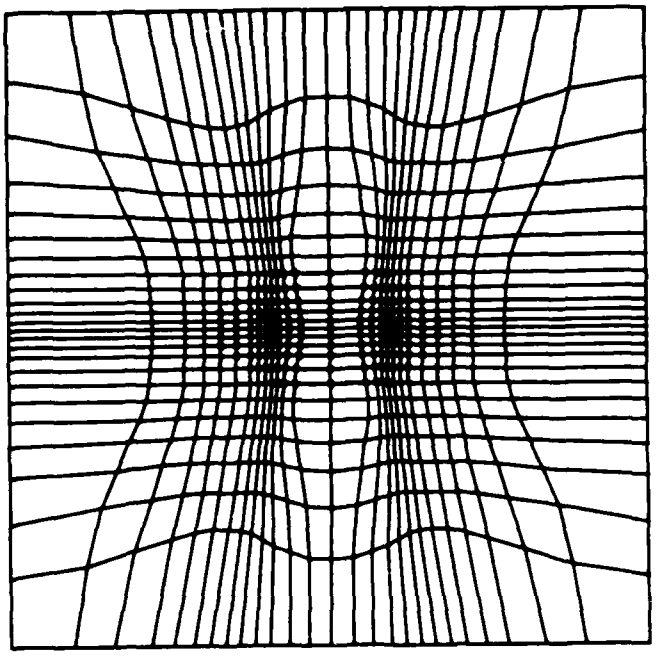


Figure 17. Mesh of Example 4.3 at t = 0.35, Clusters have merged into single cluster centered at the Origin.

Figure 3 showed the initial mesh for this problem, and Figure 17 shows the mesh at t = 0.35, just as the clusters have collided and merged. From t = 0.35 to t = 0.9, the single cluster stays centered at the origin so the mesh does not move during this time. At t = 0.9 the cones have passed completely through one another, and Figure 18 shows the separation of the error clusters and the movement of the mesh toward the boundaries. Figure 19 shows the mesh at t = 1.3. The cones and error clusters have reached the domain boundary and no further movement of the mesh will take place as the cones exit the domain.

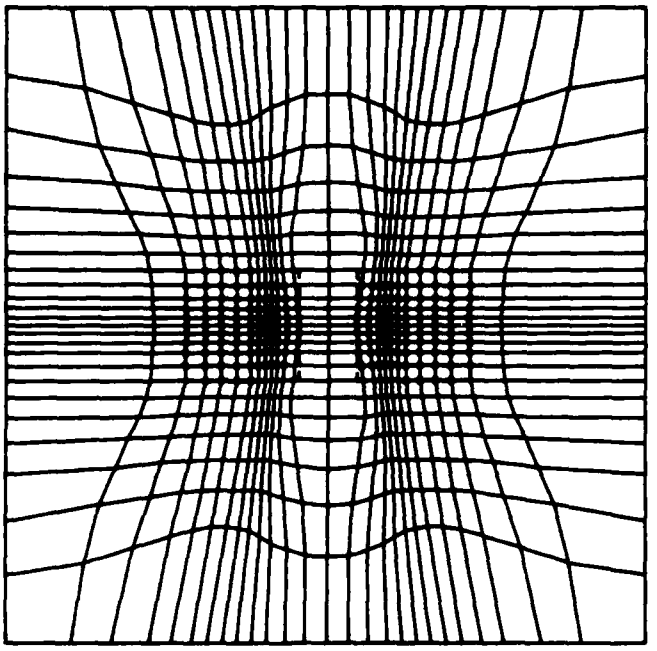


Figure 18. Mesh of Example 4.3 at t = 0.9, Clusters are separating and moving toward the Domain Boundaries.

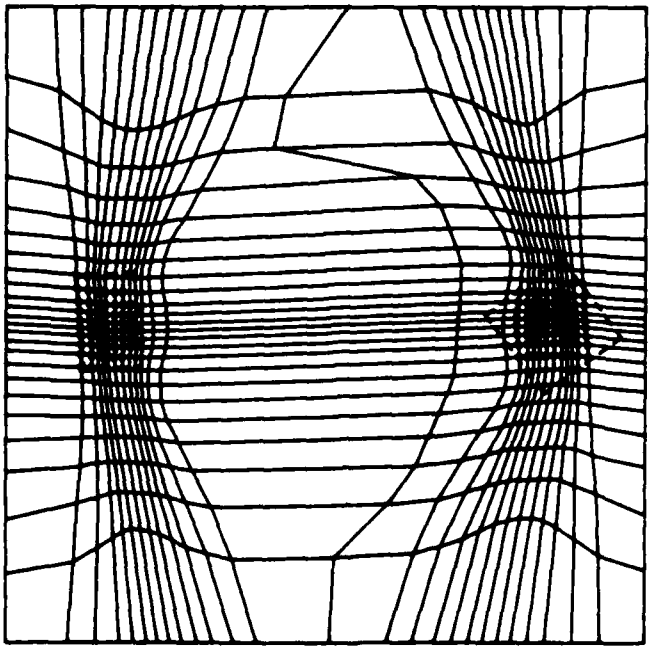


Figure 19. Mesh of Example 4.3 when Clusters reach the Domain Boundaries.

5. DISCUSSION AND CONCLUSIONS

We have described a general two-dimensional mesh moving technique based on the nodes following error propagation that is determined from the movement of clusters nodes with significantly high error. This mesh moving was tested on linear hyperbolic problems having solutions with large gradients. Even though mesh moving in two dimensions is difficult, we are encouraged by these initial results. The mesh moving algorithm was able to control the error rotation of the rotating cone in Example 4.1 and the merging and separating of error regions in Example 4.3. The distortion of the mesh in Example 4.2 showed the need for static rezoning when such distortions occur.

We are investigating ways to improve the efficiency, reliability, and robustness of the algorithm. Possible improvements include: not clustering at every time step and letting the mesh move at a constant velocity for several time steps, efficiently testing for mesh tangling or distortion, using a better solver for hyperbolic equations such as the monotonic schemes of Osher [22], vanLeer [27], or Engquist [12], and using better error estimates. We intend to show the flexibility of the mesh mover by implementing it with a finite element solver for parabolic problems.

Finally, we intend to implement local mesh refinement in the algorithm. We believe that with such a combination algorithm efficiency, accuracy, and robustness can be increased.

REFERENCES

1. BABUSKA, I., CHANDRA, J., AND FLAHERTY, J. E. (Eds.) *Adaptive Computational Methods for Partial Differtial Equations*, SIAM, Philadelphia, 1983.

2. BELL, J. B. AND SHUBIN, G. R., 'An Adaptive Grid Finite Difference Method for Conservation Laws,' *J. Comput. Phys.*, Vol. 52, pp. 569-591, 1983.

3. BERGER, M., 'Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,' Ph.D Thesis, Computer Science Department, Stanford University, 1982.

4. BERGER, M., 'Data Structures for Adaptive Mesh Refinement,' Babuska, I., Chandra, J., Flaherty, J. E.,(eds), *Adaptive Computational Methods for Partial Differential Equations*, SIAM, Philadelphia, 1983.

5. BIETERMAN, M. AND BABUSKA, I., 'The Finite Element Method for Parabolic Equations, I. A Posteriori Error Estimation,' *Numer. Math.*, Vol 40, pp. 339-371, 1982.

6. BIETERMAN, M. AND BABUSKA, I., 'The Finite Element Method for Parabolic Equations, II. A Posteriori Error Estimation and Adaptive Approach,' *Numer. Math.*, Vol 40, pp. 373-406, 1982.

7. BRACKBILL, J. U. AND SALTZMAN, J. S., 'Adaptive Zoning for

Singular Problems in Two Dimensions,' *Journal of Comp. Physics*, Vol 46, pp. 342-368, 1982.

8. COYLE, M., FLAHERTY, J. E., AND LUDWIG, R., 'On the Stability of Mesh Equidistributing Strategies for Time Dependent Partial Differential Equations,' submitted to *Journal of Comp. Physics*.

9. DAVIS, S. AND FLAHERTY, J. E., 'An Adaptive Finite Element Method for Initial-Boundary Value Problems for Partial Differential Equations,' *Siam J. Sci. Stat. Comput.*, Vol. 3, pp. 6-27,1982.

10. DREW, D. A. AND FLAHERTY, J. E., 'Adaptive Finite Element Methods and the Numerical Solution of Shear Band Problems,' to appear in M. Gurtin (ed.) *Phase Transitions and Material Instabilities in Solids*, Academic Press, 1984.

11. DWYER, H. A., 'Grid Adaption for Problems with Separation, Cell Reynolds number, Shock-Boundary Layer Interaction, and Accuracy,' AIAA Paper No. 83-0449, AIAA Twenty First Aerospace Sciences Meeting, 1983.

12. ENGQUIST, B. AND OSHER, S., 'One-sided Difference Approximations for Nonlinear Conservation Laws,' *Math. Comp.*, Vol 36, pp. 321-351, 1981.

13. FLAHERTY, J. E. AND MOORE, P. K., 'An Adaptive Local

Refinement Finite Element Method for Parabolic Partial Differential Equations,' to appear in *Proc. Conf. Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Lisbon, 1984.

14. FLAHERTY, J. E., COYLE, J. M., LUDWIG, R., AND DAVIS, S. F., 'Adaptive Finite Element Methods for Parabolic Partial Differential Equations,' *Adaptive Commputational Methods for Partial Differential Equations*, Babuska, I., Chandra, J., and Flaherty, J. E. (Eds), SIAM, Philadelphia, 1983.

15. GELINAS, R. J., DOSS, S. K., and MILLER, K., 'The Moving Finite Element Method: Application to General Partial Differential Equations with Large Gradients,' *Journal of Computational Physics*, Vol 40, 1981.

16. GANNON, D., 'Self Adaptive Methods for Parabolic Partial Differential Equations,' Department of Computer Science, University of Illinois at Urbana-Champaign, 1980.

17. GOTTLIEB, D. AND ORSZAG, S., *Numerical Analysis of Spectral Methods Theory and Applications*, SIAM, 1977.

18. HARTEN, A. AND HYMAN, J. M., 'Self-Adjusting Grid Methods for One Dimensional Hyperbolic Conservation Laws,' *J. of Comput Phys.*, Vol 50, pp. 235-269, 1983.

19. HYMAN, J. M., 'Adaptive Moving Mesh Methods for Partial

Differential Equations,' Los Alamos National Laboratory report LA-UR-82-3690.

20. MILLER, K. AND MILLER, R. N., 'Moving Finite Elements I,' *SIAM J. Num. Anal.*, Vol 18, pp. 1019-1032, 1981.

21. MILLER, K., Moving Finite Elements, II, *Siam J. Num Anal.*, Vol 18, pp. 1033-1057, 1981.

22. OSHER, S. AND CHAKRAVARTHY, S., 'Upwind Schemes and Boundary Conditions with Applications to Euler Equations in General Geometries,' *J. Comput. Phys*, Vol 50, pp. 447-481, 1983.

23. RAI, M. AND ANDERSON, D., 'The Use of Adaptive Grids in Conjunction with Shock Capturing Methods,' AIAA Paper 81-1012, June 1981.

24. RAI, M. AND ANDERSON, D., 'Application of Adaptive Grids in Fluid Flow Problems with Asymptotic Solutions,' AIAA Paper 81-0114, January 1981.

25. RAI, M. AND ANDERSON, D., 'Grid Evolution in Time Asymptotic Problems,' *Journal of Computational Physics*, Vol 43, October 1981.

26. SALTZMAN, J. S. AND BRACKBILL, J., 'Applications and Generalizations of Variational Methods for Generating Adaptive Meshes,' *Numerical Grid Generation*, 1982.

27. VAN LEER, B., 'Computational Methods for Ideal Compressible Flow,' NASA Report 172180, 1983.

28. ZIENKIEWICZ, O. C., KELLY, D. W., GAGO, J., AND BABUSKA, I., 'Hierarchical Finite Element Approaches, Error Estimates and Adaptive Refinement', *Proc. MAFELAP 1981*, April 1981.

TECHNICAL REPORT INTERNAL DISTRIBUTION LIST

	NO. OF COPIES
CHIEF, DEVELOPMENT ENGINEERING BRANCH	
ATTN: SMCAR-LCB-D	1
-DA	1
-DP	1
-DR	1
-DS (SYSTEMS)	1
-DS (ICAS GROUP)	1
-DC	1
CHIEF, ENGINEERING SUPPORT BRANCH	
ATTN: SMCAR-LCB-S	1
-SE	1
CHIEF, RESEARCH BRANCH	
ATTN: SMCAR-LCB-R	2
-R (ELLEN FOGARTY)	1
-RA	1
-RM	2
-RP	1
-RT	1
TECHNICAL LIBRARY ATTN: SMCAR-LCB-TL	5
TECHNICAL PUBLICATIONS & EDITING UNIT ATTN: SMCAR-LCB-TL	2
DIRECTOR, OPERATIONS DIRECTORATE	1
DIRECTOR, PROCUREMENT DIRECTORATE	1
DIRECTOR, PRODUCT ASSURANCE DIRECTORATE	1

NOTE: PLEASE NOTIFY DIRECTOR, BENET WEAPONS LABORATORY, ATTN: SMCAR-LCB-TL, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST

	NO. OF COPIES
ASST SEC OF THE ARMY RESEARCH & DEVELOPMENT ATTN: DEP FOR SCI & TECH THE PENTAGON WASHINGTON, D.C. 20315	1
COMMANDER DEFENSE TECHNICAL INFO CENTER ATTN: DTIC-DDA CAMERON STATION ALEXANDRIA, VA 22314	12
COMMANDER US ARMY MAT DEV & READ COMD ATTN: DRCDE-SG 5001 EISENHOWER AVE ALEXANDRIA, VA 22333	1
COMMANDER ARMAMENT RES & DEV CTR US ARMY AMCCOM ATTN: SMCAR-LC	1
SMCAR-LCE	1
SMCAR-LCM (BLDG 321)	1
SMCAR-LCS	1
SMCAR-LCU	1
SMCAR-LCW	1
SMCAR-SCM-O (PLASTICS TECH EVAL CTR, BLDG. 351N)	1
SMCAR-TSS (STINFO) DOVER, NJ 07801	2
DIRECTOR BALLISTICS RESEARCH LABORATORY ATTN: AMXBR-TSB-S (STINFO) ABERDEEN PROVING GROUND, MD 21005	1
MATERIEL SYSTEMS ANALYSIS ACTV ATTN: DRXSY-MP ABERDEEN PROVING GROUND, MD 21005	1
COMMANDER US ARMY AMCCOM ATTN: SMCAR-ESP-L ROCK ISLAND, IL 61299	1
COMMANDER ROCK ISLAND ARSENAL ATTN: SMCRI-ENM (MAT SCI DIV) ROCK ISLAND, IL 61299	1
DIRECTOR US ARMY INDUSTRIAL BASE ENG ACTV ATTN: DRXIB-M ROCK ISLAND, IL 61299	1
COMMANDER US ARMY TANK-AUTMV R&D COMD ATTN: TECH LIB - DRSTA-TSL WARREN, MI 48090	1
COMMANDER US ARMY TANK-AUTMV COMD ATTN: DRSTA-RC WARREN, MI 48090	1
COMMANDER US MILITARY ACADEMY ATTN: CHMN, MECH ENGR DEPT WEST POINT, NY 10996	1
US ARMY MISSILE COMD REDSTONE SCIENTIFIC INFO CTR ATTN: DOCUMENTS SECT, BLDG. 4484 REDSTONE ARSENAL, AL 35898	2
COMMANDER US ARMY FGN SCIENCE & TECH CTR ATTN: DRXST-SD 220 7TH STREET, N.E. CHARLOTTESVILLE, VA 22901	1

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH AND DEVELOPMENT CENTER, US ARMY AMCCOM, ATTN: BENET WEAPONS LABORATORY, SMCAR-LCB-TL, WATERVLIET, NY 12189, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST (CONT'D)

	NO. OF COPIES
COMMANDER US ARMY MATERIALS & MECHANICS RESEARCH CENTER ATTN: TECH LIB - DRXMR-PL WATERTOWN, MA 01272	2
COMMANDER US ARMY RESEARCH OFFICE ATTN: CHIEF, IPO P.O. BOX 12211 RESEARCH TRIANGLE PARK, NC 27709	1
COMMANDER US ARMY HARRY DIAMOND LAB ATTN: TECH LIB 2800 POWDER MILL ROAD ADELPHIA, MD 20783	1
COMMANDER NAVAL SURFACE WEAPONS CTR ATTN: TECHNICAL LIBRARY CODE X212 DAHLGREN, VA 22448	1
DIRECTOR US NAVAL RESEARCH LAB ATTN: DIR, MECH DIV CODE 26-27, (DOC LIB) WASHINGTON, D.C. 20375	1 1
COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/DLJ AFATL/DLJG EGLIN AFB, FL 32542	1 1
METALS & CERAMICS INFO CTR BATTELLE COLUMBUS LAB 505 KING AVENUE COLUMBUS, OH 43201	1

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH AND DEVELOPMENT CENTER, US ARMY AMCCOM, ATTN: BENET WEAPONS LABORATORY, SMCAR-LCB-TL, WATERVLIET, NY 12189, OF ANY ADDRESS CHANGES.

END

DTIC

5-86